

DATA TRANSMISSION SEQUENCING METHOD ASSOCIATED WITH BRIDGING DEVICE AND APPLICATION SYSTEM

5 CROSS-REFERENCE TO RELATED APPLICATION

This application claims the priority benefit of Taiwan application serial no.
90113546, filed June 5, 2001.

BACKGROUND OF THE INVENTION

10 Field of Invention

[0001] The present invention relates to a data transmission sequencing method.
More particularly, the present invention relates to a bridging device between data buses
and a method for ensuring proper data transmission sequence.

15 Description of Related Art

[0002] In computer systems, a bridging device is a common piece of hardware.
Various data buses are connected together by bridging devices so that data can be sent
freely to various devices via the buses. For speeding up the data movement, the
implement of posted write buffers are essential. Furthermore, to ensure transmission
20 accuracy, data must be transmitted in proper sequence.

[0003] Fig. 1 is a sketch of a conventional bridging device and associated data
buses. As shown in Fig. 1, the bridging device 180 of computer system is connected to
at least one primary bus 100 and a secondary bus 110. A typical read/write cycle for
this bridging device 180 includes a primary-to-secondary write operation 170 (PS_w), a

primary-to-secondary read operation 150 (PS_r), a secondary-to-primary write operation 160 (SP_w) and a secondary-to-primary read operation 140 (SP_r). In a conventional computer system, the bridging device uses a type of simple data transmission sequence to process data transfer. To process a read operation, the bridging device first checks if write data within write buffers 120 or 130 have already been transmitted. In other words, if a read operation is following a few write operations, the read operation can only be executed when all previous write data stored in the write buffer 120 or 130 have been transmitted.

[0004] Fig. 2 is a diagram showing the read/write cycles of the various buses connected to a conventional bridging device. As shown in Figs. 1 and 2, when the secondary bus executes two write operations to transmit write data to the primary bus, a pair of posted write cycles SP_w[0] and SP_w[1] are issued on the secondary data bus. In the meantime, a primary bus to secondary bus data read request issues on the primary bus. Hence, a PS_r_a (address phase of PS_r) issues on the primary bus. To achieve proper data transmission sequence in a conventional bridging device, the bridging device suppresses subsequent actions of the secondary bus (in other words, SP_w or SP_r). Moreover, action of the PS_r must wait until all write data within posted write buffer 130 are sent to the primary bus. In other words, PS_r can be transmitted to the secondary data bus only after the posted write cycles SP_w[0] and SP_w[1] on the primary bus are issued. Consequently, the first idle time on the secondary bus can be seen.

[0005] When the read operation is transmitted to the secondary data bus, or in other words, timing the cycle PS_r is issued on the secondary bus, the second idle appears on the primary bus because the primary bus is waiting for the response data.

When the bridging device transmits the response data (PS_r_d : data phase of PS_r) to the primary bus, , the third idle time appears on the second bus because the secondary bus needs to wait for response data actually transmitted to the primary bus. Thereafter, secondary-to-primary bus write request is resumed. For example, posted write cycles
5 SP_w[2] and SP_w[3] are issued on the secondary bus. In brief, this type of data transmission sequencing scheme leads to three major idle times resulting in a lowering of data transmission speed.

SUMMARY OF THE INVENTION

10 [0006] Accordingly, one object of the present invention is to provide a data transmission sequencing method for enhancing the data transmission rate.

[0007] To achieve these and other advantages and in accordance with the purpose of the invention, as embodied and broadly described herein, the invention provides a method for sequencing data transmission. The method comprises the steps
15 of: recording a plurality of write data stored inside the second-to-first posted write buffer when a first-to-second bus read operation via the bridging device occurs; holding a response data obtained after executing the first-to-second bus read operation on the second bus; and continuously transmitting the response data to the first bus after transmitting the write data to the first bus.

20 [0008] This invention also provides a bridging device between a first bus and a second bus. The bridging device includes a posted write buffer for storing a plurality of write data transmitting from the second bus to the first bus. The bridging device further includes a hold buffer for holding a response data after executing a read operation on the second bus, and continuously transmitting the response data to the first

bus after transmitting the write data, which are issued before the read operation, to the first bus.

[0009] It is to be understood that both the foregoing general description and the following detailed description are exemplary, and are intended to provide further
5 explanation of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The accompanying drawings are included to provide a further understanding of the invention, and are incorporated in and constitute a part of this
10 specification. The drawings illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention. In the drawings,

[0011] Fig. 1 is a sketch of a conventional bridging device and associated data buses;

[0012] Fig. 2 is a diagram showing the read/write cycles of the various buses
15 connected to a conventional bridging device;

[0013] Fig. 3 is a sketch of a bridging device and associated data buses according to one preferred embodiment of this invention; and

[0014] Fig. 4 is a diagram showing the read/write cycles of the various buses connected to a bridging device according to one preferred embodiment of this invention.

20

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0015] Reference will now be made in detail to the present preferred embodiments of the invention, examples of which are illustrated in the accompanying

drawings. Wherever possible, the same reference numbers are used in the drawings and the description to refer to the same or like parts.

[0016] Fig. 3 is a sketch of a bridging device and associated data buses according to one preferred embodiment of this invention. As shown in Fig. 3, the bridging device 280 connects with a primary bus 100 and a secondary bus 110. The bridging device 280 also includes a secondary-to-primary posted write buffer (S-to-P Write Buffer) 130. When the primary bus 100 needs to execute a PS_r operation 150, no matter whether the S-to-P Write Buffer 130 stores write data or not, the bridging device 280 can execute the read operation when the secondary bus 110 is not in use. Afterwards, the secondary bus 110 is still permitted to execute the posted write cycles SP_w to the primary bus 100, i.e. to store write data in the S-to-P Write Buffer 130 inside the bridging device 280.

[0017] The bridging device 280 further includes a primary-to-secondary posted write buffer (P-to-S Write Buffer) 120. When the secondary bus 110 needs to execute a SP_r operation 140, no matter whether or not the P-to-S Write Buffer 120 stores write data, the bridging device 280 can execute the read operation when the primary bus 100 is not in use. Afterwards, the primary bus 100 is still permitted to execute posted write cycles PS_w to the secondary bus 110, i.e. to store write data in the P-to-S Write Buffer 120 inside the bridging device 280.

[0018] The bridging device 280 also includes a secondary-to-primary flush flag buffer (S-to-P flush flag buffer) 220 that corresponds with the S-to-P Write Buffer 130 and records its state information. The bridging device 280 further includes a block logic for primary-to-secondary read (block logic for P to S read) 230 that monitors state information inside the S-to-P flush flag buffer 220. The bridging device 280 also

has a hold buffer for primary-to-secondary read (hold buffer for P to S read) 240 for holding response data from the secondary bus 110 and receiving a control signal submitted from the block logic for P to S read 230. The S-to-P flush flag buffer 220 holds each state information inside the S-to-P Write Buffer 130. The block logic for P to S read 230 outputs a control signal to the hold buffer for P to S read 240 according to the state information so that whether to transmit response data to the primary bus 100 or not can be determined.

[0019] The bridging device 280 also includes a primary-to-secondary flush flag buffer (P-to-S flush flag buffer) 190 that corresponds to the P-to-S Write Buffer 120 and records its state information. The bridging device 280 further includes a block logic for secondary-to-primary read (block logic for S to P read) 200 that monitors state information inside the P-to-S flush flag buffer 190. The bridging device 280 also has a hold buffer for secondary-to-primary read (hold buffer for S to P read) 210 for holding response data from the primary bus 100 and receiving a control signal submitted from the block logic for S to P read 200. The P-to-S flush flag buffer 190 holds each state information inside the P-to-S Write Buffer 120. The block logic for S to P read 200 outputs a control signal to the hold buffer for S to P read 210 according to the state information so that whether or not to transmit response data to the secondary bus 110 can be determined.

[0020] If the P-to-S Write Buffer 120 transmits the write data to the secondary bus 110 in sequential order, the P-to-S flush flag buffer 190 and the block logic for S to P read 200 can be implemented using a counter. When a secondary-to-primary read request is issued, the counter may record the number of current write data existing inside the P-to-S Write Buffer 120. Moreover, the counter decreases by one after

transmitting each write data to the secondary bus 110. The response data stored inside the hold buffer for S to P read 210 is transmitted to the secondary bus 110 as soon as the counter returns to zero.

[0021] If the P-to-S Write Buffer 120 transmits the write data but not necessarily in sequential order, the block logic for S to P read 200 can reproduce a content of the state information stored in P-to-S flush flag buffer 190. When a secondary-to-primary read request is issued, the block logic for S to P read 200 reproduces a content of the state information stored in P-to-S flush flag buffer 190. After transmitting each write data to the secondary bus 110, the corresponding state information stored in the P-to-S flush flag buffer 190 is cleared. The response data stored inside the hold buffer for S to P read 210 is transmitted to the secondary bus 110 when the write data having the same content of state information stored in the P-to-S flush flag buffer 190 are all cleared.

[0022] Fig. 4 is a diagram showing the read/write cycles of the various buses connected to a bridging device according to one preferred embodiment of this invention. As shown in Figs. 3 and 4, if two write operations are executed from the secondary bus 110 to the primary bus 100, two posted write cycles SP-w[0] and SP_w[1] issue on the secondary bus. The write data is placed inside the S-to-P Write Buffer 130. The S-to-P flush flag buffer 220 stores two state information indicating that two write data need to be transmitted to the primary bus 100. In the meantime, a primary bus to secondary bus data read request issues on the primary bus. Hence, a PS_r_a (address phase of PS_r) issues on the primary bus.

[0023] According to this embodiment, the bridging device 280 includes the S-to-P flush flag buffer 220, the block logic for P to S read 230 and the hold buffer for P

to S read 240. Hence, after transferring the two write data from the secondary bus to the S-to-P Write Buffer 130 (during the posted write cycles SP_w[0] and SP_w[1]), read request (PS_r) is immediately issued on the secondary bus 110 and the response data then is stored in the hold buffer for P to S read 240. After the PS_r_a (address phase
5 of the PS_r) issued on the primary bus 100, two write data within the S-to-P Write Buffer 130 are transferred to the primary bus 100. The block logic for P to S read 230 monitors the state information stored in the S-to-P flush flag buffer 220. After transmitting the two write data to the primary bus 100, the block logic for P to S read 230 submits a control signal to the hold buffer for P to S read 240. Therefore, the
10 response data stored in hold buffer for P to S read 240 is directly transmitted to the primary bus 100 (during data phase of PS_r_d).

[0024] Obviously, as long as the secondary bus 110 is not in use, the secondary-to-primary bus requests can be resumed. For example, two posted write cycles SP_w[2] and SP_w[3] may issued on the secondary bus 110. In summary, the method
15 of sequencing data transmission used by the bridging device 280 can prevent idle sessions and hence boost data transmission rate while maintaining proper data transmission sequence. In addition, the bridging device may serve as a communication link between various types of data buses including PCI bus, AGP bus, DRAM bus and CPU bus.

[0025] It will be apparent to those skilled in the art that various modifications and variations can be made to the structure of the present invention without departing from the scope or spirit of the invention. In view of the foregoing, it is intended that the present invention cover modifications and variations of this invention provided they fall within the scope of the following claims and their equivalents.